Journal of Computational and Applied Mathematics 224 (2009) 514–526

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A nonmonotone conic trust region method based on line search for solving unconstrained optimization[☆]

Shao-Jian Qu[a,*], Qing-Pu Zhang[a], Yue-Ting Yang[b]

[a] *Harbin Institute of Technology, Harbin 150080, PR China*

[b] *Faculty of Science, Xi'an Jiaotong University, Xi'an 710049, PR China*

ARTICLE INFO

Article history:
Received 29 November 2007

Keywords:
Unconstrained optimization
Trust region method
Conic model
Nonmonotone technique
Line search technique

ABSTRACT

In this paper, we present a nonmonotone conic trust region method based on line search technique for unconstrained optimization. The new algorithm can be regarded as a combination of nonmonotone technique, line search technique and conic trust region method. When a trial step is not accepted, the method does not resolve the trust region subproblem but generates an iterative point whose steplength satisfies some line search condition. The function value can only be allowed to increase when trial steps are not accepted in close succession of iterations. The local and global convergence properties are proved under reasonable assumptions. Numerical experiments are conducted to compare this method with the existing methods.

1. Introduction

In this paper, the following unconstrained optimization is considered,

$$\min_{x\in R^n} f(x) \tag{1.1}$$

where $f : R^n \to R$ is twice continuously differentiable function.

Trust region methods for unconstrained optimization have been studied by many researchers [1–5]. Trust region methods are robust, can be applied to ill-conditioned problems and have strong global convergence properties. Another advantage of trust region methods is that there is no need to require the approximate Hessian matrix of the trust region subproblem to be positive definite. For problem (1.1), Nocedal and Yuan [6] showed that a trust-region trial step is always a descent direction for any approximate Hessian matrix. It is well known that for line search methods one generally has to assume the approximate Hessian matrix to be positive definite in order to ensure that the search direction is a descent direction.

In [18], we proposed a new trust region subproblem based on conic model for unconstrained optimization,

$$\begin{cases} \min c_k(s) = \dfrac{g_k^T s}{1-h_k^T s} + \dfrac{1}{2}\dfrac{s^T B_k s}{(1-h_k^T s)^2} \\ s.t. 1-h_k^T s > 0 \\ \|s\| \le \Delta_k \end{cases} \tag{1.2}$$

where $c_k(s)$ is called conic model which is an approximation to $f(x_k+s)-f(x_k)$, B_k is an approximate Hessian of f at x_k and Δ_k is the trust radius. The vector h_k is the associated vector for the colinear scaling in the kth iteration, and it is normally called

☆ This work is supported by the National Science Foundation Grant (10671057) of China.

* Corresponding author.
E-mail address: qushaojian@163.com (S.-J. Qu).

0377-0427/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2008.05.028

the horizontal vector. If $h_k = 0$, the conic model reduces to a quadratic model. Therefore, the conic model methods are the generalization of the quadratic model methods. They have several advantages. First, if the objective function has strong non-quadratic behavior or its curvature changes severely, the quadratic model methods often produce a poor prediction of the minimizer of the function. In this case, conic model approximates the objective function better than a quadratic, because it has more freedom in the model. Second, the quadratic model does not take into account the information concerning the function value in the previous iteration which is useful for algorithms. However, the conic model possesses richer interpolation information and satisfies four interpolation conditions of the function values and the gradient values at the current and the previous points. Using these rich interpolation information may improve the performance of the algorithms. Third, the initial and limited numerical results provided in [8,9] etc. show that the conic model method gives improvement over the quadratic model method. Finally, the conic model method has the similar global and local convergence properties as the quadratic model method.

Some criterion is used to decide whether or not the trial step s_k is accepted. If the trial step is not accepted, the subproblem (1.2) with a reduced trust region radius should be resolved until an acceptable step is found. Hence, the subproblem may be solved several times at an iteration and the total cost of computation for one iteration might be expensive for large scale problem.

In recent years, a variety of trust region methods have been proposed in the literatures. For example, Nocedal and Yuan [19], and Gertz [20] presented methods which combine line search technique and trust region method. When the trial step is not successful, their methods performs a line search to find a iterative point instead of resolving the subproblem. Therefore, their methods require less computation than classic trust region methods. On the other hand, Sun and Zhang [21], Chen and Sun [22], and Mo and Zhang [12] proposed a fixed steplength method for unconstrained optimization. In their approaches, without using line search, they computed the steplength by a formula at each iteration. Thus their methods might be practical in cases where the line search is expensive or hard.

Recently, nonmonotone line search techniques have been studied by many authors since Grippo et al. [10]. Many authors generalized the nonmonotone technique to trust region methods and proposed nonmonotone trust region methods [11–13]. Theoretical analysis and numerical results show that the algorithms with nonmonotone properties are more efficient than the algorithms with monotone properties. These papers indicated that the nonmonotone algorithm is efficient, especially for ill-conditioned problems.

To our knowledge, the nonmonotone trust region methods listed above are mostly based on quadratic model, but there are less nonmonotone trust region methods based on conic model [18]; and the trust region methods based on line search techniques listed above are all based on quadratic model.

In our paper, we combine the subproblem (1.2) with nonmonotone and line search techniques to propose a nonmonotone trust region method based on conic model and line search techniques. This method is different from the one in [18], since in this paper the method is based on line search, i.e. at every iteration, the subproblem (1.2) is only solved once and the solution of (1.2) is used as the search step. The local and global convergence properties of the nonmonotone trust region method based on conic model are proved under some reasonable assumptions. Finally, we report some preliminary numerical experiments and compare the performance of the new method with the performance of the method in [18]. The numerical results show that the new method performs better.

The rest of the paper is organized as follows. In Section 2, we present the nonmonotone trust region method based on conic model and line search. In Section 3, the global and local convergence properties are studied. Numerical results in Section 4 indicate that the algorithm is efficient. Concluding remarks are addressed in Section 5.

2. The algorithm

In this section, we describe a method which combines nonmonotone technique, linear search technique and conic trust region method. In each iteration, a trial step s_k is generated by solving the trust region subproblem (1.2). Then either $x_k + s_k$ is accepted as a new iteration point or the trust-region radius is reduced according to a comparison between the actual reduction of the objective function

$$ared_k(s_k) = f_{l(k)} - f(x_k + s_k) \tag{2.1}$$

and the reduction predicted by the conic model

$$pred_k(s_k) = -\frac{g_k^T s_k}{1 - h_k^T s_k} - \frac{1}{2}\frac{s_k^T B_k s_k}{(1 - h_k^T s_k)^2}. \tag{2.2}$$

i.e.,

$$r_k = \frac{ared_k(s_k)}{pred_k(s_k)}, \tag{2.3}$$

where

$$f_{l(k)} = \max_{0 \le j \le m(k)} \{f_{k-j}\}, \quad k = 1, 2, \ldots, \tag{2.4}$$

and $m(k)$ is an integer defined by

$$m(k) = \begin{cases} 0 & \text{if } r_{k-1} \geq c_2 \\ \min\{m(k-1)+1, M\} & \text{otherwise,} \end{cases} \tag{2.5}$$

where $M \geq 1$ is an integer constant and $m(0) := 0$. That is, if the reduction in the objective function is satisfactory, then we finish the current iteration by taking

$$x_{k+1} = x_k + s_k \tag{2.6}$$

and adjusting the trust-region radius; otherwise a new iterative point by $x_{k+1} = x_k + \alpha_k s_k$ is generated, where $\alpha_k := \rho^{j_k}$ is a steplength satisfying

$$f_k - f(x_k + \rho^j s_k) \geq -\delta \rho^j g_k s_k, \tag{2.7}$$

here $\delta > 0$, $\rho \in (0, 1)$ are constants and j^k is the smallest integer, $j = 0, 1, 2, \ldots$ such that the above inequality holds.

Algorithm 2.1 (*The Nonmonotone Conic Trust Region Algorithm Based on Line Search Technique for Unconstrained Optimization*). **Step 0.** Choose parameters $0 < c_2 < 1 < c_1$, $0 < c_3 < c_4 < 1 \Delta_{\min} > 0$ and $\varepsilon \geq 0$; give a starting point $x_1 \in R^n$, $B_1 \in R^{n\times n}$, $h_1 \in R^n$, an integer constant $M \geq 0$ and an initial trust-region radius $\Delta_{\min} \leq \Delta_1 < \Delta_{\max}$; set $k := 1$, $m(0) := 0$.

Step 1. If $\|g_k\| < \varepsilon$, then stop with x_k as the approximate optimal solution; otherwise go to Step 2.

Step 2. Solve the conic minimization subproblem (1.2) and let s_k be one solution of the subproblem (1.2).

Step 3. Compute $m(k)$ by (2.5). Compute $ared_k(s_k)$, $pred_k(s_k)$ and

$$r_k = \frac{ared_k(s_k)}{pred_k(s_k)}.$$

If $r_k \geq c_2$, then set

$$x_{k+1} := x_k + s_k, \tag{2.8}$$

and go to Step 1.

Step 4. Compute α_k by (2.7) and set

$$x_{k+1} := x_k + \alpha_k s_k. \tag{2.9}$$

Step 5. Set

$$\Delta_{k+1} \begin{cases} = \max[c_1 \Delta_k, \Delta_{\min}], & \text{if } r_k \geq c_2 \\ \in [c_3 \Delta_k, c_4 \Delta_k] & \text{otherwise.} \end{cases} \tag{2.10}$$

Step 6. Generate h_{k+1} and B_{k+1}; set $k := k + 1$, and go to Step 1.

Remarks. (i) For the trust-region-based methods, the main computation is spent to solve the trust-region subproblem. It is well known that solving the trust-region subproblem exactly is expensive. Hence developing approximate methods for the trust-region subproblem has been a popular research topic since 1980s and numerous algorithms have been proposed. Recently, for solving the subproblem (1.2) an efficient approximate Algorithm 4.1 of [7] has been proposed. In this paper, we will use this algorithm to solve the conic trust-region subproblem (1.2).

(ii) The method for generating α_{k+1} and B_{k+1} can be seen, for example, in [14–16]. The conditions that we assume for proving global convergence are that the matrices B_k are uniformly bounded and

$$\forall k, \quad \exists \sigma \in (0, 1) : \|h_k\| \Delta_k \leq \sigma \tag{2.11}$$

which ensures that the conic model function $\varphi_k(s)$ is bounded over the trust-region $\{s \mid \|s\| \leq \Delta_k\}$. We would like to reiterate the fact that our algorithm reduces to a quadratic model based algorithm if $h_k = 0$ for all k. Note that, under the smoothness assumptions taken in this paper, the objective function is locally convex quadratic around a local minimizer. It means that choosing $h_k \simeq 0$ asymptotically is suitable when x_k is near the minimizer.

(iii) If $M = 0$, this algorithm reduces to monotone one.

3. Global convergence

In this section, we establish the global convergence results of our algorithm given in the previous section.

Before we address some theoretical issues, we would like to make the following assumptions.

Assumption 3.1. (i) The level set $L(x_1) = \{x \in R^n | f(x) \le f(x_1)\}$ is bounded.

(ii) The sequences $\{B_k\}$ are positive definition and there exists $\mu_1 > 0$ such that

$$d^T B_k d \ge \mu_1 d^T d, \quad \forall d \in R^n \text{ and } k = 1, 2, \ldots. \tag{3.1}$$

(iii) The function f is LC^1 in R^n, i.e., there exists $\mu_2 > 0$ such that

$$\|\nabla f(x) - \nabla f(y)\| \le \mu_2 \|x - y\|, \quad \forall x, y \in R^n. \tag{3.2}$$

For simplicity, we define two index sets as follows:

$$I = \{k : r_k \ge c_2\} \quad \text{and} \quad J = \{k : r_k < c_2\}. \tag{3.3}$$

Define

$$\tau^* = \frac{g_k^T g_k}{C} \tag{3.4}$$

where $C = g_k^T B_k g_k - h_k^T g_k g_k^T g_k$. τ_k is defined as

$$\tau_k := \begin{cases} \tau^*, & \text{if } |\tau^*| \cdot \|g_k\| \le \Delta_k \text{ and } \tau^* > 0 \\ \dfrac{\Delta_k}{\|g_k\|}, & \text{otherwise.} \end{cases} \tag{3.5}$$

The following theorem implies that the search direction $s_k = -\tau_k g_k$ can guarantee the sufficient predicted reduction.

Lemma 3.1. *Suppose that h_k and Δ_k are all bounded above, τ_k is defined as (3.5), then*

$$\omega(\tau_k) = f_k - c_k(-\tau_k g_k) \ge \delta_1 \|g_k\| \cdot \min\left\{\frac{\|g_k\|}{\|B_k\|}, \|s_k\|\right\} \tag{3.6}$$

where $\delta_1 = \min\{1, \frac{1}{2(1+\Delta_{\max})M_h}\}$, $\Delta_{\max} \ge \Delta_k$, $M_h \ge \|h_k\|$, $\forall k$.

Proof. a. If $|\tau^*| \cdot \|g_k\| \le \Delta_k$ and $\tau^* > 0$, $\tau_k = \tau^*$. Hence

$$\omega(\tau_k) = f_k - c_k(-\tau_k g_k) = \frac{\tau^* g_k^T g_k}{1 + \tau^* h_k^T g_k} - \frac{(\tau^*)^2 g_k^T B_k g_k}{2(1 + \tau^* h_k^T g_k)^2}.$$

By (3.4), we get $g_k^T g_k = \frac{\tau^* g_k^T B_k g_k}{1+\tau^* h_k^T g_k}$.

So

$$\begin{aligned}\omega(\tau_k) &= \frac{\tau^* g_k^T g_k}{1 + \tau^* h_k^T g_k} - \frac{\tau^* g_k^T g_k}{2(1 + \tau^* h_k^T g_k)} \\ &= \frac{\tau^* g_k^T g_k}{2(1 + \tau^* h_k^T g_k)} = \frac{(g_k^T g_k)^2}{2 g_k^T B_k g_k} \\ &\ge \frac{(g_k^T g_k)^2}{2\|g_k\|^2 \|B_k\|} \ge \frac{g_k^T g_k}{2\|B_k\|}.\end{aligned} \tag{3.7}$$

b. If $\tau^* \cdot \|g_k\| > \Delta_k$, $\tau_k = \frac{\Delta_k}{\|g_k\|}$. Obviously, $\tau^* = \frac{g_k^T g_k}{C} > \frac{\Delta_k}{\|g_k\|}$, then we have,

$$g_k^T g_k \|g_k\| + \Delta_k h_k^T g_k g_k^T g_k > \Delta_k g_k^T B_k g_k \tag{3.8}$$

and

$$C = g_k^T B_k g_k - h_k^T g_k g_k^T g_k > 0. \tag{3.9}$$

Therefore

$$\begin{aligned}\omega(\tau_k) &= \frac{\tau_k g_k^T g_k}{1 + \tau_k h_k^T g_k} - \frac{\tau_k^2 g_k^T B_k g_k}{2(1 + \tau_k h_k^T g_k)^2} \\ &= \frac{\Delta_k g_k^T g_k}{\|g_k\| + \Delta_k \alpha_k^T g_k} - \frac{\Delta_k^2 g_k^T B_k g_k}{2(\|g_k\| + \Delta_k h_k^T g_k)^2}\end{aligned}$$

$$
\begin{aligned}
&> \frac{\Delta_k g_k^T g_k}{\|g_k\| + \Delta_k h_k^T g_k} - \frac{\Delta_k g_k^T g_k}{2(\|g_k\| + \Delta_k h_k^T g_k)} \\
&= \frac{\Delta_k g_k^T g_k}{2(\|g_k\| + \Delta_k h_k^T g_k)} \\
&\geq \frac{\Delta_k \|g_k\|}{2(1 + \Delta_{\max} M_h)}
\end{aligned}
\tag{3.10}
$$

where the first inequality comes from (3.8); the second inequality comes from $\|g_k\| + \Delta_k h_k^T g_k \leq \|g_k\|(1 + \Delta_{\max} M_h)$.

c. If $\tau^* < 0$ or τ^* does not exist, $\tau_k = \frac{\Delta_k}{\|g_k\|}$.
If $\tau^* < 0$, we have $C = g_k^T B_k g_k - h_k^T g_k g_k^T g_k < 0$.
If τ^* does not exist, we have $C = 0$.
Thus in both cases we always have $C \leq 0$, this together with B_k positive definite implies that $h_k^T g_k > 0$.
Therefore,

$$
\begin{aligned}
\omega(\tau_k) &= \frac{\tau_k g_k^T g_k}{1 + \tau_k h_k^T g_k} - \frac{\tau_k^2 g_k^T B_k g_k}{2(1 + \tau_k h_k^T g_k)^2} \\
&= \frac{\Delta_k g_k^T g_k}{\|g_k\| + \Delta_k h_k^T g_k} - \frac{\Delta_k^2 g_k^T B_k g_k}{2(\|g_k\| + \Delta_k h_k^T g_k)^2} \\
&> \frac{\Delta_k g_k^T g_k}{\|g_k\| + \Delta_k h_k^T g_k} - \frac{\Delta_k g_k^T g_k}{2(\|g_k\| + \Delta_k h_k^T g_k)} \\
&= \frac{\Delta_k g_k^T g_k}{2(\|g_k\| + \Delta_k h_k^T g_k)} \\
&\geq \frac{\Delta_k \|g_k\|}{2(1 + \Delta_{\max} M_h)}
\end{aligned}
\tag{3.11}
$$

where the first inequality comes from $h_k^T g_k > 0$.

Then by combining a., b., c., we have that

$$
\omega(\tau_k) \geq \|g_k\| \cdot \min\left\{ \frac{\|g_k\|}{\|B_k\|}, \frac{\Delta_k}{2(1 + \Delta_{\max} M_h)} \right\}. \tag{3.12}
$$

Let $\delta = \min\{1, \frac{1}{2(1+\Delta_{\max})M_h}\}$. Because $\|s_k\| \leq \Delta_k$, so from (3.12),

$$
\omega(\tau_k) = f_k - c_k(-\tau_k g_k) \geq \delta \|g_k\| \cdot \min\left\{ \frac{\|g_k\|}{\|B_k\|}, \|s_k\| \right\}. \quad \blacksquare
$$

Based on the above theorem, we have that the solution s_k of subproblem (1.2) can guarantee the sufficient predicted reduction.

Theorem 3.2. *Suppose that h_k and Δ_k are all bounded above. Then*

$$
pred_k(s_k) \geq \delta_1 \|g_k\| \min\left[\Delta_k, \frac{\|g_k\|}{\|B_k\|} \right], \quad \forall k, \tag{3.13}
$$

where s_k is the solution to (1.2), δ_1 is defined as in Lemma 3.1.

Proof. According to the definitions of s_k, we have that

$$
\varphi_k(0) - \varphi_k(s_k) \geq \varphi_k(0) - \varphi_k(-\tau_k g_k). \tag{3.14}
$$

This together with Lemma 3.1 implies that this theorem is true.

Under assumption (2.11) and Assumption 3.1(ii), the above theorem implies that

$$
s_k^T g_k \leq -\delta_2 \|g_k\| \min\left[\Delta_k, \frac{\|g_k\|}{\|B_k\|} \right], \tag{3.15}
$$

which implies that there must be some α_k such that (2.7) holds, where $\delta_2 = (1 - \sigma)\delta_1$. Furthermore the above theorem shows that s_k generated by (1.2) is one descent direction and then there must exist α_k such that (2.7) holds. Furthermore, the following lemma can be obtained by Lemma 3.2 of [18].

Lemma 3.3. *Suppose that* $\{x_k\} \subseteq L(x_1)$, (2.11) *and* Assumption 3.1 *hold, then there exists one positive constant* δ_3 *such that*

$$|f_k - f(x_k + s_k) - pred_k(s_k)| \le \delta_3 \|s_k\|^2, \quad \forall k. \tag{3.16}$$

The following theorem shows that the line search (2.7) must stop finitely.

Theorem 3.4. *Let* $\{x_k\}$ *be the sequence generated by* Algorithm 2.1. *If* Assumption 3.1 *and* (2.11) *hold, then there exits* $\eta > 0$ *such that*

$$f(x_k + \alpha_k s_k) - f_{l(k)} \le \eta g_k^T s_k, \quad \forall k \in J. \tag{3.17}$$

Proof. First, we will show that there must be $\bar{\alpha} \in (0, 1)$, such that $\alpha_k \ge \bar{\alpha}$, $\forall k$. Consider the function

$$\phi_k(\alpha) = f(x_k + \alpha s_k) - f(x_k) - \delta \alpha g_k^T s_k. \tag{3.18}$$

By Assumption 3.1, $f(x_k + \alpha s_k)$ is bounded below, and $g_k^T s_k < 0$ immediately follows from (3.15) if algorithm does not terminate finite. Therefore $\lim_{\alpha\to\infty} \phi_k(\alpha) = \infty$. The first and second derivatives of ϕ_k at zero are

$$\phi_k' = (1 - \delta) g_k^T s_k \quad \text{and} \quad \phi_k'' = s_k^T \nabla^2 f(x_k) s_k.$$

Since $g_k^T s_k$ is negative, then so is $\phi_k'(0)$. Because $\phi_k(0)$ equals zero and $\phi_k'(0)$ is negative, $\phi_k(\alpha)$ is negative for all sufficiently small positive $\alpha \in (0, 1]$. Because there are $\alpha \in (0, 1]$ with $\phi_k(\alpha) < 0$ but $\phi_k(\alpha)$ is continuous and $\lim_{\alpha\to\infty} \phi_k(\alpha) = \infty$, there is a $\beta > 0$ with $\phi_k(\beta) = 0$. Without loss of generality, we assume that $\beta \in (0, 1]$. Let α^* be the global minimizer of $\phi_k(\alpha)$ in $[0, \beta]$. The minimum value cannot occur at the end points because $\phi_k(0) = \phi_k(\beta) = 0$, but there are $\alpha \in [0, \beta]$ with $\phi_k(\alpha) < 0$. Thus α^* is also a local minimizer of $\phi_k(\alpha)$, and $\phi_k(\alpha^*) < 0$.

Let α' by any local minimizer of $\phi_k(\alpha)$ with $\phi_k(\alpha') < 0$. Then

$$f(x_k + \alpha' s_k) - f(x_k) - \delta \alpha' g_k^T s_k < 0,$$

so

$$f(x_k + \alpha' s_k) - f(x_k) < \delta \alpha' g_k^T s_k.$$

Therefore α' satisfies (2.7) and the existence of an appropriate $\bar{\alpha} \in (0, 1)$ then follows immediately from the existence of a local minimizer and the continuity of $\phi_k(\alpha)$.

Now we prove (3.17). The definition of $f_{l(k)}$ implies that $f_k \le f_{l(k)}$ for all k. Then the conclusion follows immediately from the above conclusion and $g_k^T s_k \le 0$ with $\eta = \delta\bar{\alpha}$. ■

This theorem also indicates that steplength α_k satisfying (2.7) can be found in finite iterations.

The following theorem shows that the algorithm is well defined.

Theorem 3.5. *Suppose that* (2.11) *and* Assumption 3.1 *hold. Then* Algorithm 2.1 *is well defined.*

Proof. The conclusion follows from both Theorem 3.2 and the proof of Theorem 3.4. ■

The following lemma shows that the sequence $\{x_k\}$ generated by Algorithm 2.1 is contained in the level set $L(x_1)$.

Lemma 3.6. *Let* $\{x_k\}$ *generated by* Algorithm 2.1. *If* Assumption 3.1 *and* (2.11) *hold, then* $\{f_{l(k)}\}$ *is monotonically decreasing. Furthermore,* $\{x_k\} \subset L(x_1)$.

Proof. We firstly show that the sequence $\{f_{l(k)}\}$ is monotonically decreasing, i.e.,

$$f_{l(k+1)} \le f_{l(k)} \tag{3.19}$$

for all k.

For $k \in I$, it follows from $r_k \ge c_2$, Algorithm 2.1 and Lemma 3.1 that $f(x_{k+1}) \le f_{l(k)}$. On the other hand, by $r_k \ge c_2$ and (2.5), we have $m(k + 1) = 0$. Hence, $f(x_{k+1}) = f_{l(k+1)}$ and then (3.19) holds for $k \in I$.

Suppose that $k \in J$. It follows from $g_k^T s_k \le 0$ and the line search that

$$f_{k+1} \le f_{l(k)}. \tag{3.20}$$

If $m(k + 1) = 0$, then $f_{k+1} = f_{l(k+1)}$ and the above inequality implies that (3.19) holds. If $m(k + 1) > 0$, then $m(k + 1) \le m(k) + 1$. By the definition of $f_{l(k)}$ and (3.20), we have (3.19) holds. Then (3.19) holds for all k.

Now, we prove the last conclusion. The definition of $f_{l(k)}$ and Step 5 of Algorithm 1 imply that $f_k \le f_{l(k)}$ and $f_{l(1)} = f_1$. By (3.19), we have $f_{l(k)} \le f_{l(1)}$, $\forall k$. Hence, $\{x_k\} \subset L(x_1)$. ■

The next theorem shows that if Algorithm 2.1 does not stop finite, then there will be sufficient real reduction.

Theorem 3.7. *Suppose that Assumption 3.1 and (2.11) hold, and there exists $\varepsilon > 0$ such that $\|g_k\| \geq \varepsilon$ for all k. Then there exists a constant $\delta_4 > 0$ such that*

$$f_{l(k)} - f_{k+1} \geq \delta_4 \min\left\{\Delta_k, \frac{\varepsilon}{\|B_k\|}\right\} \tag{3.21}$$

holds for all k.

Proof. If $k \in I$ or equivalently $r_k \geq c_2$, it follows from Theorem 3.2 that

$$f_{l(k)} - f_{k+1} \geq c_2 pred_k(s_k) \geq c_2\delta_1\varepsilon \min\left\{\Delta_k, \frac{\varepsilon}{\|B_k\|}\right\}. \tag{3.22}$$

If $k \in J$, from Theorem 3.4 and (3.15), it follows that

$$f_{l(k)} - f_{k+1} \geq \delta_2\eta\varepsilon \min\left\{\Delta_k, \frac{\varepsilon}{\|B_k\|}\right\}. \tag{3.23}$$

Thus (3.22) and (3.23) imply that (3.21) holds for all k with $\delta_4 = \min\{c_2\delta_1\varepsilon, \delta_3\eta\varepsilon\}$. ■

Lemma 3.8. *Suppose that Assumption 3.1 and (2.11) hold, and there exists $\varepsilon > 0$ such that $\|g_k\| \geq \varepsilon$ for all k. Then there exists $\nu \in (0, 1)$ such that*

$$\lim_{k\to\infty} \min\left\{\nu\Delta_k, \frac{\varepsilon}{Z_k}\right\} = 0 \tag{3.24}$$

where $Z_k = 1 + \max_{1\leq i\leq k} \|B_k\|$.

Proof. We define S to be the set of integer k such that $m(k) = 0$. Let $\{i_j : j = 1, 2, \ldots\}$ be a infinite set of integer which contains S and satisfies

$$1 \leq i_{j+1} - i_j < M + 1 \tag{3.25}$$

for all j, and

$$i_{j+1} - i_j = M + 1 \tag{3.26}$$

if $i_{j+1} \bar{\in} S$. Note that $i_1 = 1$ for $m(1) = 0$.

Next, we show that inequality

$$f_{l(i_j)} - f_{l(i_{j+1})} \geq \delta_4 \min\left\{\frac{\Delta_{i_{j+1}}}{c_1}, \frac{\varepsilon}{Z_{i_{j+1}}}\right\} \tag{3.27}$$

holds for all j. We consider two cases separately.

Case 1. $i_{j+1} - i_j = 1$. By (3.26) and $M \geq 1$, we have $i_{j+1} \in S$. This implies $m(i_{j+1}) = 0$. Then from the definition of $f_{l(k)}$, we obtain $f_{l(i_{j+1})} = f_{i_{j+1}}$. On the other hand, $m(i_{j+1}) = 0$, (2.5) and (2.10) imply that

$$\Delta_{i_j} \geq \frac{\Delta_{i_{j+1}}}{c_1}. \tag{3.28}$$

From Theorem 3.7, (3.28) and the monotonicity of $\{Z_k\}$, it follows that (3.27) holds.

Cases 2. $i_{j+1} - i_j > 1$. For $s = 1, 2, \ldots, i_{j+1} - i_j - 1$, by the definition of $\{i_j\}$, we have $m(i_j + s) > 0$. It follows from (2.5) that $r_{i_j+s} < c_2$. Then from (2.10), we have

$$\Delta_{i_j} \geq \Delta_{i_j+1} \geq \Delta_{i_j+2} \geq \cdots \geq \Delta_{i_{j+1}-1} \geq \Delta_{i_{j+1}} \tag{3.29}$$

if $r_{i_j} < c_2$, or

$$c_1\Delta_{i_j} \geq \Delta_{i_j+1} \geq \Delta_{i_j+2} \geq \cdots \geq \Delta_{i_{j+1}-1} \geq \Delta_{i_{j+1}} \tag{3.30}$$

if $r_{i_j} \geq c_2$.

If $i_{j+1} \in S$, then $m(i_{j+1}) = 0$ and $f_{l(i_{j+1})} = f_{i_{j+1}}$. From Theorem 3.7, we have

$$f_{l(i_{j+1}-1)} - f_{i_{j+1}} \geq \delta_4 \min\left\{\Delta_{i_{j+1}-1}, \frac{\varepsilon}{Z_{i_{j+1}-1}}\right\}. \tag{3.31}$$

From (3.21), it follows that $f_{l(i_j)} \geq f_{l(i_{j+1}-1)}$. Thus from (3.29)–(3.31) and the monotonicity of $\{Z_k\}$ we obtain

$$f_{l(i_j)} - f_{l(i_{j+1})} \geq \delta_4 \min\left\{\Delta_{i_{j+1}}, \frac{\varepsilon}{Z_{i_{j+1}}}\right\}. \tag{3.32}$$

Now, we assume that $i_{j+1} \bar{\in} S$. For $s = 1, 2, \ldots, i_{j+1} - i_j - 1$, by $c_1 > 1$, Theorem 3.7, (3.29) and (3.30) and the definition of Z_k, we have

$$\begin{aligned} f_{l(i_j+s)} &\geq f_{i_j+s+1} + \delta_4 \min\left\{\Delta_{i_j+s}, \frac{\varepsilon}{B_{i_j+s}}\right\} \\ &\geq f_{i_j+s+1} + \delta_4 \min\left\{\frac{\Delta_{i_{j+1}}}{c_1}, \frac{\varepsilon}{Z_{i_{j+1}}}\right\}. \end{aligned} \tag{3.33}$$

By (3.19) and (3.26), it follows that

$$\begin{aligned} f_{l(i_j)} &\geq \max\{f_{i_j+s+1} : 0 \leq s \leq M\} + \delta_4 \min\left\{\frac{\Delta_{i_{j+1}}}{c_1}, \frac{\varepsilon}{Z_{i_{j+1}}}\right\} \\ &\geq f_{l(i_{j+1})} + \delta_4 \min\left\{\frac{\Delta_{i_{j+1}}}{c_1}, \frac{\varepsilon}{Z_{i_{j+1}}}\right\}, \end{aligned} \tag{3.34}$$

where the last inequality follows from the definition of $f_{l(k)}$ and $m(i_j + 1) \leq M$. Since $c_1 \geq 1$, from (3.32) and (3.34), we see that (3.27) holds.

Now, we prove the conclusion of this lemma holds, i.e., (3.24) holds. By Lemma 3.6 and Assumption 3.1, the sequence $\{f_{l(i_j)}\}$ is convergent. Then summing inequalities (3.27), we obtain

$$\sum_{j=1}^{\infty} \min\left\{\frac{\Delta_{i_j+1}}{c_1}, \frac{\varepsilon}{Z_{i_{j+1}}}\right\} \leq f_{l(i_1)} - \lim_{j\to\infty} f_{l(i_{j+1})} < +\infty. \tag{3.35}$$

From (3.25), (3.29), (3.30) and the monotonicity of $\{Z_k\}$, we get

$$\begin{aligned} \sum_{k=1}^{\infty} \min\left\{\frac{\Delta_k}{c_1^2}, \frac{\varepsilon}{Z_k}\right\} &= \sum_{j=1}^{\infty} \sum_{s=i_j}^{i_{j+1}-1} \min\left\{\frac{\Delta_s}{c_1^2}, \frac{\varepsilon}{Z_s}\right\} \\ &\leq \sum_{j=1}^{\infty} (i_{j+1} - i_j) \min\left\{\frac{\Delta_{i_j}}{c_1}, \frac{\varepsilon}{Z_{i_j}}\right\} \\ &< (M+1) \min\left\{\frac{\Delta_{i_j}}{c_1}, \frac{\varepsilon}{Z_{i_j}}\right\} < +\infty. \end{aligned}$$

This implies that (3.24) holds with $v = \frac{1}{c_1^2} \in (0, 1)$. ■

Lemma 3.9. *Suppose that Assumption 3.1 and (2.11) hold, and there exists $\varepsilon > 0$ such that $\|g_k\| \geq \varepsilon$ for all k. Then the following inequality*

$$\|s_k\| \geq \min\{1, \delta_1 \varepsilon (1 - c_2)(1 - \sigma)^2\}/Z_k \tag{3.36}$$

holds for $k \in J$ sufficiently large.

Proof. Notice that Algorithm 2.1 ensures that $\|s_k\| \leq \Delta_k$. If the inequality $\|s_k\| > \delta_1 \varepsilon (1 - c_2)(1 - \sigma)^2/(2\mu)$ holds for sufficiently large $k \in J$, it follows from the conclusion of Lemma 3.8 that

$$1/Z_k \leq \delta_1 \varepsilon (1 - c_2)(1 - \sigma)^2/(2\mu)$$

holds for sufficiently large $k \in J$. Then

$$\|s_k\| > 1/Z_k \tag{3.37}$$

holds for sufficiently large $k \in J$.

Now assume that $\|s_k\| \le \delta_1\varepsilon(1-c_2)(1-\sigma)^2/(2\mu)$ for $k \in J$. Since $r_k < c_2$ for $k \in J$, by the definition of r_k and $f_{l(k)} \ge f_k$, we have

$$f_k - f(x_k + s_k) < c_2 pred_k(s_k). \tag{3.38}$$

Using Mean-value theorem and Assumption 3.1(iii), we obtain

$$\begin{aligned} f_k - f(x_k + s_k) &= -g(\bar{x}_k)^T s_k = -g_k^T s_k + (g_k - g(\bar{x}_k))^T s_k \\ &\ge -g_k^T s_k - \mu\|s_k\|^2 \ge -g_k^T s_k - \delta_1\varepsilon(1-c_2)(1-\sigma)^2\|s_k\|/2 \end{aligned} \tag{3.39}$$

where $\bar{x}_k \in [x_k, x_k + s_k]$. It follows from (2.11), (3.38) and (3.39) that

$$\begin{aligned} (1-c_2)\left(\frac{g_k^T s_k}{1-h_k^T s} + \delta_1\varepsilon\|s_k\|/2\right) &\ge \left(1 - \frac{1}{1-h_k^T s_k}\right) g_k^T s_k + c_2\frac{s_k^T B_k s_k}{2(1-h_k^T s_k)^2} \\ &> c_2\frac{s_k^T B_k s_k}{2(1-h_k^T s_k)^2}. \end{aligned} \tag{3.40}$$

From $\|g_k\| \ge \varepsilon$ and Theorem 3.2, we have

$$pred_k(s_k) \ge \delta_1\varepsilon \min\left[\Delta_k, \frac{\varepsilon}{\|B_k\|}\right]. \tag{3.41}$$

Then, it follows from $-\frac{s_k^T B_k s_k}{(1-h_k^T s_k)^2} \le \frac{\|s_k\|^2\|B_k\|}{(1-\sigma)^2}$, (3.40) and (3.41) that

$$\|s_k\|^2\|B_k\| \ge \delta_1\varepsilon(1-c_2)(1-\sigma)^2 \min\{\|s_k\|, 2\varepsilon/\|B_k\| - \|s_k\|\}. \tag{3.42}$$

If $\|s_k\| > 2\varepsilon/\|B_k\| - \|s_k\|$, it holds

$$\|s_k\| > \varepsilon/\|B_k\|. \tag{3.43}$$

Otherwise, we have

$$\|B_k\|\|s_k\| \ge \delta_1\varepsilon(1-c_2)(1-\sigma)^2. \tag{3.44}$$

Therefore from the definition of Z_k, (3.37), (3.43) and (3.44), it follows that (3.36) holds for sufficiently large $k \in J$. ∎

Lemma 3.10. *Suppose that Assumption 3.1 and (2.11) hold, and there exists $\varepsilon > 0$ such that $\|g_k\| \ge \varepsilon$ for all k. Then the following inequality*

$$\Delta_k \ge c_3 \min\{1, \delta_1\varepsilon(1-c_2)(1-\sigma)^2\}/Z_k \tag{3.45}$$

holds for sufficiently large k.

Proof. If J is a finite set, there exists a positive constant $\bar{\Delta}$ such that $\Delta_k \ge \bar{\Delta}$ for all k, then Lemma 3.8 implies that $\lim_{k\to\infty} 1/Z_k = 0$, hence (3.45) holds for all large k. Now assume that J is an infinite set. By Lemma 3.9, there exists a $\bar{k} \in J$ such that (3.36) holds for $k \in J$ and $k \ge \bar{k}$. For any $k \in I$ and $k \ge \bar{k}$, let $\hat{k} = \max\{j : j \in J \text{ and } j \le k\}$. The definition of $\hat{k}$ implies that

$$\|s_{\hat{k}}\| \ge \min\{1, \delta_1\epsilon(1-c_2)(1-\sigma)^2\}/Z_{\hat{k}} \tag{3.46}$$

and

$$\hat{k} + s \in I \tag{3.47}$$

for all $s = 1, 2, \ldots, k - \hat{k}$. Moreover, (3.47) implies that $r_{\hat{k}+s} \ge c_2$, for all $s = 1, 2, \ldots, k - \hat{k}$. From this and (2.10), we have

$$\Delta_{\hat{k}+1} \ge \Delta_{\hat{k}+2} \ge \cdots \ge \Delta_k. \tag{3.48}$$

On the other hand, from (2.10), we have $\Delta_{\hat{k}+1} \ge \Delta_{\hat{k}}$ if $r_{\hat{k}} \ge c_2$, or $c_3\|s_{\hat{k}}\| \le \Delta_{\hat{k}+1}$ if $r_{\hat{k}} < c_2$. Since $c_3 \in (0, 1)$ and Algorithm 2.1 ensures $\|s_k\| \le \Delta_k$ for all k, then

$$c_3\|s_{\hat{k}}\| \le \Delta_{\hat{k}+1}. \tag{3.49}$$

By the monotonicity of $\{Z_k\}$, (3.46), (3.48) and (3.49), we see that (3.45) holds for $k \in I$ and $k \ge \bar{k}$. ∎

Now we prove the global convergence of Algorithm 2.1.

Theorem 3.11. *If Assumption 3.1 and (2.11) hold, and $\{B_k\}$ satisfies*

$$\sum_{k=1}^{\infty} 1/Z_k = \infty \tag{3.50}$$

where $Z_k = 1 + \max_{1\le j\le k} \|B_k\|$, then the sequence $\{x_k\}$ generated by Algorithm 2.1 satisfies

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{3.51}$$

Proof. If (3.51) does not hold, there is a constant $\varepsilon > 0$ such that $\|g_k\| \ge \varepsilon$ for all k. From Lemma 3.10, there exists a integer $\bar{k}$ such that

$$\min\{\Delta_k, \epsilon/Z_k\} \ge \delta_5/Z_k \tag{3.52}$$

holds for all $k \ge \bar{k}$, where $\delta_5 = \min\{\varepsilon, \delta_1\epsilon(1-c_2)(1-\sigma)^2\}$.

Let k be any integer such that $k \ge \bar{k}$. From Theorem 3.7 and (3.52), the following inequality

$$f_{l(k+1)} \ge f_{k+s+1} + \delta_4 \min\{\Delta_{k+s}, \varepsilon/\|B_{k+s}\|\} \ge f_{k+s+1} + \delta_4\delta_5/Z_{k+s} \tag{3.53}$$

holds for $s = 0, 1, \dots, M$. From Lemma 3.8 and the monotonicity of $\{Z_k\}$, it follows that

$$\begin{aligned} f_{l(k)} &\ge \max\{f_{k+s+1} : 0 \le s \le M\} + \delta_4\delta_5/Z_{k+M+1} \\ &\ge f_{l(k+M+1)} + \delta_4\delta_5/Z_{k+M+1}, \end{aligned} \tag{3.54}$$

where the last inequality follows from the definition of $f_{l(k)}$. From Assumption 3.1 and Lemma 3.8, $\{f_{l(k)}\}$ is monotonically decreasing and convergent. Combining with (3.49), we obtain

$$\begin{aligned} \sum_{k\ge\bar{k}} 1/Z_{k+M+1} &\le (1/\delta_4\delta_5)\sum_{k\ge\bar{k}}(f_{l(k)} - f_{l(k+M+1)}) \\ &\le (1/\delta_4\delta_5)\sum_{k\ge\bar{k}}\sum_{s=0}^{M}(f_{l(k+s)} - f_{l(k+s+1)}) \\ &\le (1/\delta_4\delta_5)\sum_{k\ge 1}(f_{l(k)} - f_{l(k+1)}) < \infty. \end{aligned} \tag{3.55}$$

This contradicts (3.50). Therefore (3.51) is true. ■

4. Superlinear convergence

In order to explore the superlinear convergence we give the following assumptions.

Assumption 4.1. (i) The sequence $\{x_k\}$ generated by Algorithm 2.1 converges to a stationary point x^*, i.e.,

$$\lim_{k\to\infty} x_k = x^* \quad \text{and} \quad \lim_{k\to\infty} \|g_k\| = \|g^*\| = 0. \tag{4.1}$$

(ii) If

$$\frac{\|B_k^{-1}g_k\|}{1 - g_k^T B_k^{-1} h_k} \le \Delta_k, \tag{4.2}$$

then

$$s_k = -\frac{B_k^{-1}g_k}{1 - g_k^T B_k^{-1} h_k}. \tag{4.3}$$

The following lemma shows that for sufficiently large k, the iteration step s_k will be eventually defined as (4.3).

Lemma 4.1. *Suppose that (2.11), Assumptions 3.1 and 4.1 hold, then after finite iterations s_k must be defined as (4.3).*

Proof. Define

$$K = \left\{ k \;\middle|\; \frac{\|B_k^{-1}g_k\|}{1 - g_k^T B_k^{-1} h_k} > \Delta_k \right\}. \tag{4.4}$$

Now we will prove that the set K is finite. By (2.11), Assumption 3.1(ii) and 4.1(i), there exists $\delta_6 > 0$ such that

$$\|g_k\| \geq \delta_6 \Delta_k \geq \delta_6 \|s_k\|, \quad \forall k \in K. \tag{4.5}$$

By Lemma 3.3 and $\|s_k\| \to 0$, we have that

$$|f_k - f_{k+1} - pred_k(s_k)| = o(\|s_k\|^2). \tag{4.6}$$

By contradiction, if K is infinite, then by Assumption 4.1(i), we have that

$$\lim_{k\to\infty} \Delta_k = 0, \tag{4.7}$$

which together with $f_{l(k)} \geq f_k$ implies that the inequality

$$c_2 > r_k = \frac{f_{l(k)} - f_{k+1}}{pred_k(s_k)} \geq \frac{f_k - f_{k+1}}{pred_k(s_k)} \tag{4.8}$$

holds for all sufficiently large k. On the other hand, by Theorem 3.2, (4.5)–(4.7), there exists $\delta_7 > 0$ such that

$$\left|\frac{f_k - f_{k+1}}{pred_k(s_k)} - 1\right| \leq \frac{o(\|s_k\|^2)}{\delta_7 \|s_k\|^2} \to 0, \quad k \to \infty, \tag{4.9}$$

which means that $\frac{f_k - f_{k+1}}{pred_k(s_k)} \geq c_2$ for sufficiently large k, a contradiction to (4.8). Then we complete the proof. ■

Based on the above lemma, we can established the superlinear convergence result for Algorithm 2.1.

Theorem 4.2. *Suppose that (2.11), Assumptions 3.1 and 4.1 hold. If $\nabla^2 f(x^*)$ is positive definite and*

$$\lim_{k\to\infty} \frac{\|[B_k - \nabla^2 f(x^*)]s_k\|}{\|s_k\|} = 0, \tag{4.10}$$

then the sequence $\{x_k\}$ converges to x^* superlinearly.

Proof. By Lemma 3.12, we have that for large enough k, $\frac{\|B_k^{-1} g_k\|}{1 - g_k^T B_k^{-1} h_k} \leq \Delta_k$. Then according to Assumption 4.1(ii), for large enough k, $s_k = \frac{B_k^{-1} g_k}{1 - g_k^T B_k^{-1} h_k}$. In the following proof without loss generality, we assume that s_k is defined as (4.3) for all k. In the follows we will prove that $r_k \geq c_2$, for sufficiently large k.

Let $x_{k+1} = x_k + \lambda_k s_k$, where

$$\lambda_{k+1} = \begin{cases} 1, & \text{if } r_k \geq c_2 \\ \alpha_k, & \text{otherwise} \end{cases} \tag{4.11}$$

and α_k satisfies (2.7). According to the proof of Lemma 3.3, it follows that $\alpha_k \geq \bar{\alpha}$, $\forall k$, where $\bar{\alpha}$ is defined as in the proof of Lemma 3.3. This together with (4.11) means that

$$\lambda_k \geq \bar{\alpha} \quad \text{and} \quad \|x_{k+1} - x_k\| \geq \bar{\alpha}\|s_k\|, \forall k, \tag{4.12}$$

which together with Assumption 4.1(i) implies that

$$\|s_k\| \to 0, \quad k \to \infty.$$

It follows Assumption 4.1(ii) that

$$f_k - f(x_k + s_k) = -g_k^T s_k - \frac{1}{2} s_k^T \nabla^2 f(x^*) s_k + o(\|s_k\|^2). \tag{4.13}$$

By (2.11) and Assumption 4.1(i), there exists $\delta_8 > 0$ such that

$$pred_k(s_k) \geq \delta_8 s_k^T B_k s_k. \tag{4.14}$$

It follows (2.11), Assumption 3.1(ii), Assumption 4.1(i), (4.13) and (4.14) that there exist $\delta_9 > 0$ and $\delta_{10} > 0$, such that

$$\left|\frac{f_k - f(x_k + s_k)}{pred_k(s_k)} - 1\right| \leq \delta_9 \frac{\|[B_k - \nabla^2 f(x^*)]s_k\|}{\|s_k\|} + \delta_{10}\|g_k\|^2 + \frac{o(\|s_k\|^2)}{s_k^T B_k s_k}. \tag{4.15}$$

This together with Assumptions 3.1(ii) and 4.1(i) implies that

$$\frac{f_k - f(x_k + s_k)}{pred_k(s_k)} \geq c_2$$

Table 1
Test results

Pro.	Algorithm 2.1 n	ITR	NF	Algorithm in [18] ITR	NF
Biggs exp 6	6	154	210	476	513
Trigonometric	10^2	77	79	84	104
Extended Pow.	42	61	72	65	83
	124	82	94	70	107
	282	68	81	66	124
Extended ros.	20	79	83	Failed	
	40	85	91	Failed	
Discrete bou.	30	44	52	43	87
	55	43	62	67	118
	200	79	84	214	320
Penalty I	10	19	20	13	44
	10^2	24	31	21	56

holds for sufficiently large k. Then by $f_{l(k)} \geq f_k$ and the definition of r_k, we have that

$$r_k \geq c_2$$

holds for all sufficiently large k, i.e.,

$$x_{k+1} = x_k + s_k = x_k - \frac{B_k^{-1} g_k}{1 - g_k^T B_k^{-1} h_k}$$

holds for all sufficiently large k. So similar to the proof of Theorem 5.5.1 in [5], we can prove that Algorithm 2.1 is superlinear convergence. ■

5. Numerical experiments

In this part, we will carry out numerical experiments for the Algorithm 2.1 and compare it with the performance of the method in [18]. All programs are written in C^{++}, numerical test in PC, CPU Main Frequency 1.43G, EMS 256M, run circumstance VC++6.0, numeric type double float. The parameters in algorithm are:

$$c_1 = 1.5, \quad c_2 = 0.1, \quad c_3 = 0.5, \quad c_4 = 0.7, \quad \Delta_{\max} = 50, \quad \Delta_{\min} = \Delta_0 = 5, \quad B_0 = I.$$

The convergence criterion

$$\|g_k\| \leq 10^{-6} \quad \text{or} \quad f(x_{k-1}) - f(x_k) \leq 10^{-6} \max\{0.1, |f(x_{k-1})|\}$$

is used for the termination test; that is, when one of the two conditions is satisfied, computation stop. We also set a maximum iteration number of 500 to terminate the computation when this limit is reached.

The test problems come from [17] with fewer than 282 variables. Table 1 lists the number of iterations used for the algorithms. The column headings in this table 'ITR' and 'NF' stand for the number of iterations and the number of the valuations of functions f_k, respectively. 'Failed' means that algorithms fail to terminate at a stationary point of the problem within 500 iterations.

The results in Table 1 show that in most cases, Algorithm 2.1 performs better than the method in [18]. This means that Algorithm 2.1 is competitive.

6. Conclusions

A new trust region method is proposed for solving unconstrained optimizations. The new algorithm can be regarded as a combination of nonmonotone technique, line search technique and conic trust region method. Note that the new method is different from the classic conic trust region method, when a trial step is not accepted, the new method does not resolve the trust region subproblem but generates an iterative point whose steplength satisfies some line search condition. The function value can only be allowed to increase when trial steps are not accepted in close succession of iterations. The local and global convergence properties are proved under reasonable assumptions. Numerical experiments are conducted to compare this method with the existing methods and results show that new method is competitive.

References

[1] R. Fletcher, Practical Method of Optimization, Unconstrained Optimization, vol. 1, John Wiley, New York, 1980.
[2] M.J.D. Powell, On the global convergence of trust region algorithms for unconstrained optimization, Mathematical Programming 29 (1984) 297–303.
[3] G.A. Schultz, R.B. Schnabel, R.H. Byrd, A family of trust region based algorithms for unconstrained minimization with strong global convergence, SIAM Journal of Numerical Analysis 22 (1985) 47–67.

[4] Y. Yuan, On the convergence of trust region algorithms, Mathematica Numerica Sinica 16 (1996) 333–346.
[5] Y. Yuan, W. Sun, Optimization Theory and methods, Science Press, Beijing, China, 1997.
[6] J. Nocedal, Y. Yuan, Combining trust region and line search techniques, Report NAM 07, Department of EECS, Northwestern University, 1991.
[7] Y. Ji, S.J. Qu, Y.J. Wang, H.M. Li, A conic trust-region method for optimization with nonlinear equality and inequality constrains via active-set strategy, Applied Mathematics and Computation 183 (2006) 217–231.
[8] S. Di, W. Sun, Trust region method for conic model to solve unconstrained optimization problems, Optimization Methods and Software 6 (1996) 237–263.
[9] Q. Ni, S. Hu, A new derivative free algorithm based on conic interpolation model, Technical report, Faculty of Science, Nanjing University of Aeronautics and Astronautics, 2001.
[10] L. Grippo, F. Lampariello, S. Lucidi, A nonmonotone line search technique for Newton's Method, SIAM Journal on Numerical Analysis 23 (1986) 707–716.
[11] J.L. Zhang, X.S. Zhang, A Nonmonotone adaptive trust region method and its convergence, Computers and Mathematics with Applications 45 (2003) 1469–1477.
[12] J.T. Mo, K.C. Zhang, Z.X. Wei, A nonmonotone trust region method for unconstrained optimization, Applied Mathematics and Computation 171 (2005) 371–384.
[13] Z.W. Chen, J.Y. Han, D.C. Xu, A nonmonotone trust region method for nonlinear programming with simple bound constraints, Applied Mathematics and Optimization 43 (2001) 63–85.
[14] W.C. Davidon, Conic approximation and collinear scaling for optimizers, SIAM Journal on Numerical Analysis 17 (1980) 268–281.
[15] D.C. Sorensen, The q-superlinear convergence of a collinear scaling algorithm for unconstrained optimization, SIAM Journal of Numerical Analysis 17 (1980) 84–114.
[16] K.A. Ariyawansa, Deriving collinear scaling algorithms as extensions of quasi-Newton methods and the local convergence of DFP and BFGS related collinear scaling algorithm, Mathematical Programming 49 (1990) 23–48.
[17] J.J. Moré, B.S. Garbow, K.E. Hilstron, Testing unconstrained optimization software, ACM Transaction on Mathematical Software 7 (1981) 17–41.
[18] S.J. Qu, K.C. Zhang, J. Zhang, A nonmonotone trust region method of conic model for unconstrained optimization, Journal of Computation and Applied Mathematics doi:10.1016/j.cam.2007.07.038.
[19] J. Nocedal, Y. Yuan, Combining trust region and linear search techniques, in: Y. Yuan (Ed.), Advances in Nonlinear Programming, Kluwer Academic Publishers, Dordrecht, 1996, pp. 153–175.
[20] E.M. Gertz, Combination trust-region line search methods for unconstrained optimization, University of California, San Diego, 1999.
[21] J. Sun, J.P. Zhang, Global convergence of conjugate gradient methods without line search, Annals of Operational Research 103 (2001) 161–173.
[22] X.D. Chenand, J. Sun, Global convergence of a two-parameter family of conjugate gradient methods without line search, Journal of Computation and Applied Mathematics 146 (2002) 37–45.